LU03.A04 - Webformulare: JSON-Daten lesen, bearbeiten, einfügen und löschen

1. Bestandteile eine node-Servers kennen.
2. Basis-Server ausführen können.
3. Daten anzeigen, verändern und schreiben können.

Rahmenbedingungen

- Sozialform: individual
- Hilfsmittel: OpenBooks
- Zeit: 30 Minutes
- Erwartetes Resultat: Node-Server, der verschiedene abgefragte Daten am Monitor/Browser bearbeitet und ausgibt.

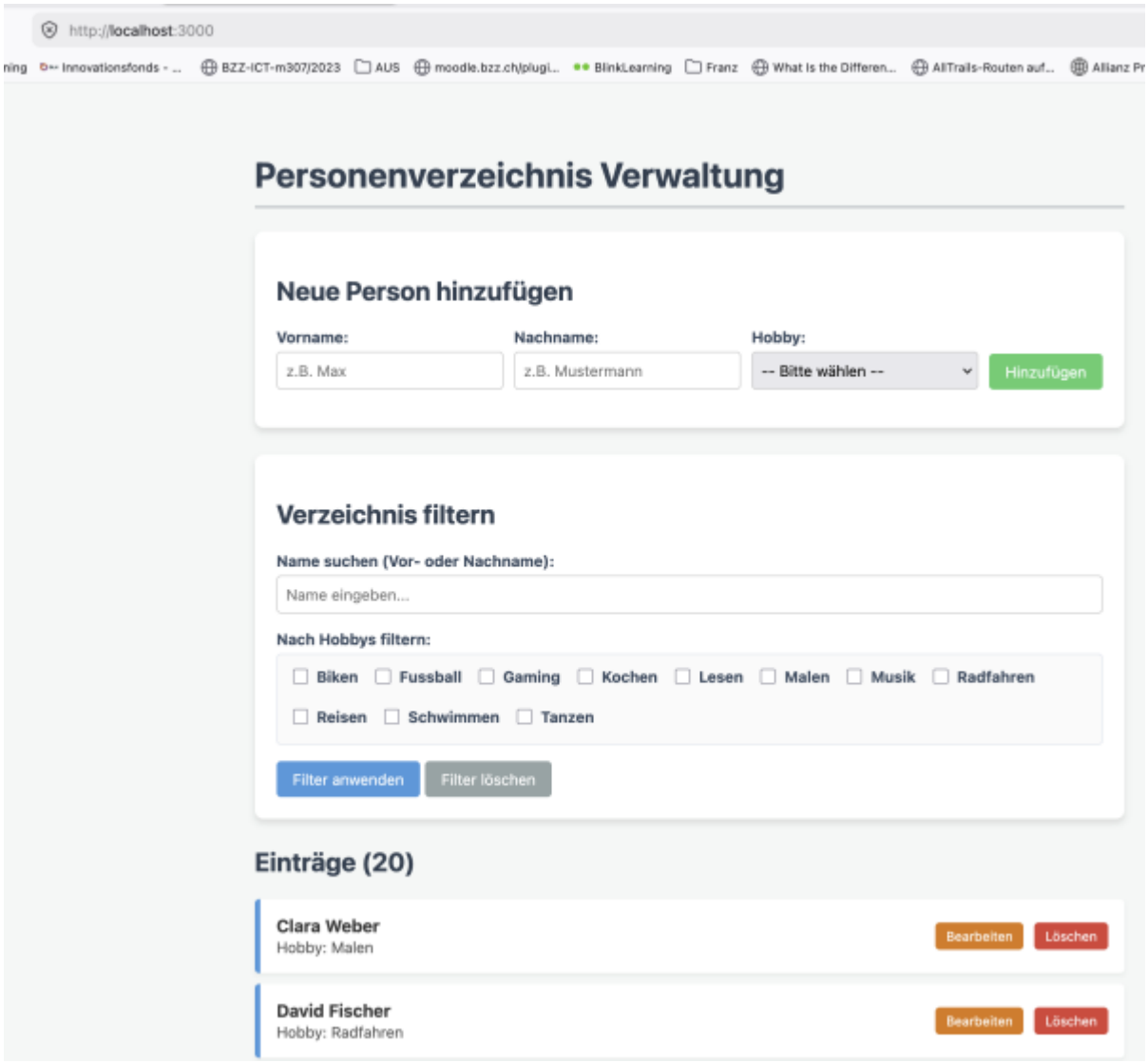
Ausgangslage

Um Rahmen des Web-Entwicklungsprojekts „Personenverzeichnis“ soll eine bestehende Node.js-Applikation mit Express und EJS (Embedded JavaScript) finalisiert und in Betrieb genommen werden. Bisher blockieren fehlerhafte Dateipfade, unvollständige Routen-Variablen sowie Caching-Probleme im Firefox-Browser das fehlerfreie Hinzufügen, Bearbeiten und Löschen von Datensätzen. Ihre Aufgabe ist es, die Ordnerstruktur zu bereinigen, die Server-Logik zu vervollständigen und die Benutzeroberfläche mit einem dynamischen Hobby-Dropdown sowie optimierten Buttons auszustatten. Der lokale Prototyp muss anschliessend auf Port 3000 fehlerfrei und stabil laufen.

Auftrag

Installieren Sie die Ihnen zur Verfügung gestellten Files wie unten abgebildet und lassen Sie die *Mini-Applikation* laufen. Legen Sie Personen an, filtern diese und löschen auch Datensätze.

Wenn alles korrekt gemacht wurde, sieht Ihre Applkation ungefähr wie folgt aus:



Benötigte File und Ablage-Struktur

Die nachfolgenden Files müssen wir folgt im VS Studio Code abgelegt werden.

Benötigten Files

Stellen Sie also sicher, dass diese den nachfolgenden Aufbau haben.

```
▼ a04 (Projekt-Wurzelverzeichnis)
  ▼ public
    ▼ css
        style.css
  ▼ views
      index.ejs
    data.json
    server.js
```

Arbeitsschritte

1. Order & Files

- Laden Sie die Files herunter
- Importieren Sie diese ins VS Studio Code

2. Projektabhängigkeiten

- Bevor das Skript starten kann, müssen die benötigten Node-Pakete installiert werden.

```
npm init -y
```

**. 3. Express ** Die benötigten Module installieren: Express (für den Server) und EJS (für die Templates) müssen im Projekt registriert werden:

```
npm install express ejs
```

** 4. Schreibberechtigung** Damit wir die JSON-Datei auch beschreiben können, müssen wir hier die Berechtigung anpassen.

** 4.1 Winddows-Powershell**

```
# Entfernt das Schreibschutz-Attribut (R = Read-Only) von der Datei
attrib -r data.json
```

4.2 MacOS Gehen Sie im Finder auf das JSON-File –> Get Info –>

2. Server

Erstellen Sie ein weiteres File mit dem Titel *server.js* und kopieren den nachfolgenden Code rein.

```
const express = require('express');
const app = express();
const PORT = 3000;

// Middleware, damit Express JSON-Daten im Body lesen kann
app.use(express.json());

// Unser "Datenbank"-Ersatz
let users = ["Alice", "Bob"];

// --- DEINE AUFGABE AB HIER ---

// 1. GET ROUTE: Gib das 'users' Array zurück
app.get('/users', (req, res) => {
    // CODE HIER
});
```

```
// 2. POST ROUTE: Füge einen neuen User hinzu
app.post('/users', (req, res) => {
    const newUser = req.body.name; // Wir erwarten { "name": "DeinName" }
    if (newUser) {
        // CODE HIER: User zum Array hinzufügen
        res.status(201).send(`User ${newUser} wurde hinzugefügt.`);
    } else {
        res.status(400).send("Fehler: Kein Name angegeben!");
    }
});

// --- ENDE DER AUFGABE ---

app.listen(PORT, () => {
    console.log(`Server läuft auf http://localhost:${PORT}`);
});
```

Lösungen

LU03.L01

Volkan Demir

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m307/learningunits/lu03/aufgaben/04?rev=1779801157

Last update: **2026/05/26 15:12**